

CASE STUDY

Recall

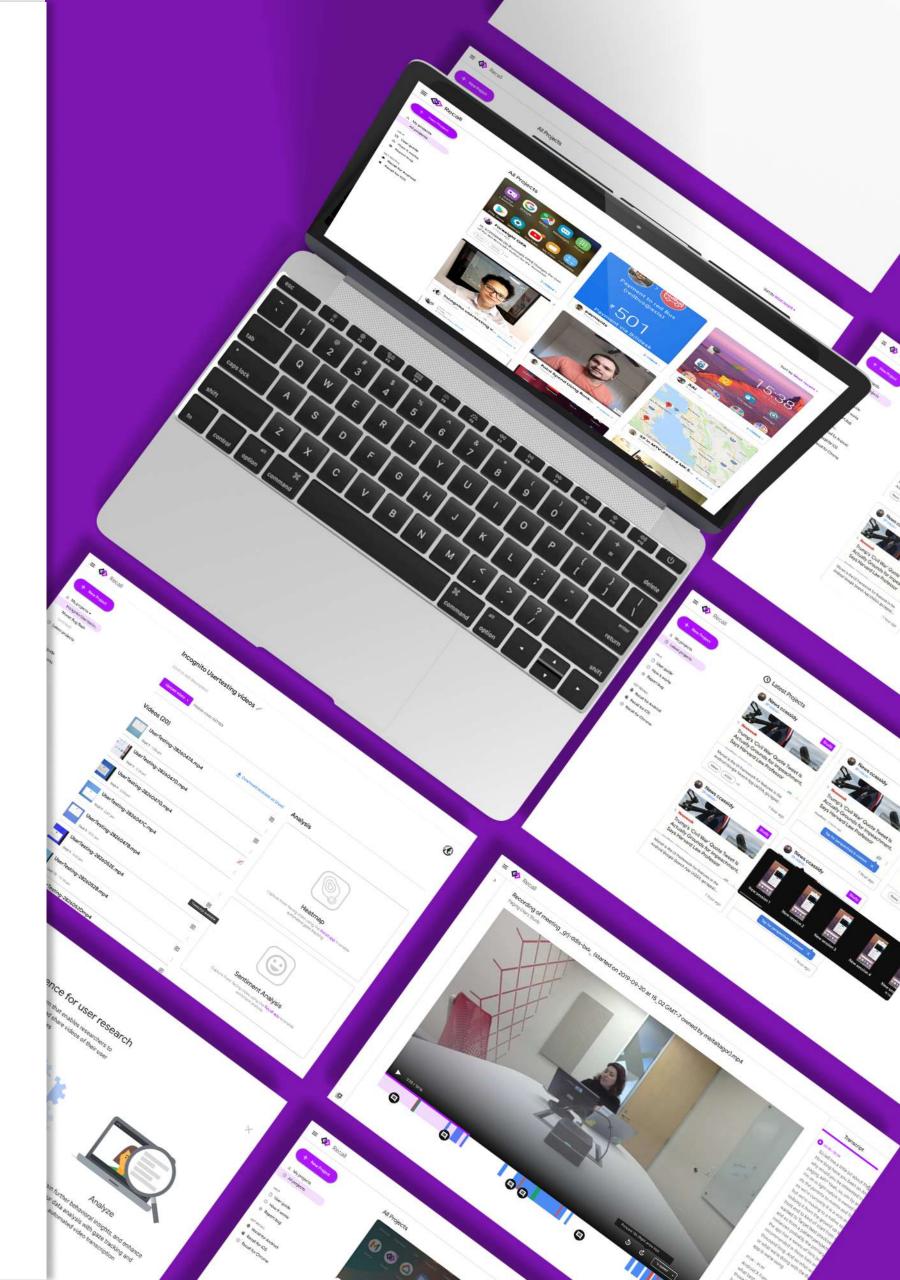
A video platform at Google that enables UX researchers to record, upload, analyze, and share videos of their user studies.

- Product Management
- Illustration

UX Research

Development

UX Design



ABOUT RECALL

Recall is an internal web app used by thousands of UX researchers at Google to record, store, and analyze user studies.

THE PROBLEM

Recall wasn't built with design in mind. Pages and flows were disorderly. The app needed a design foundation.

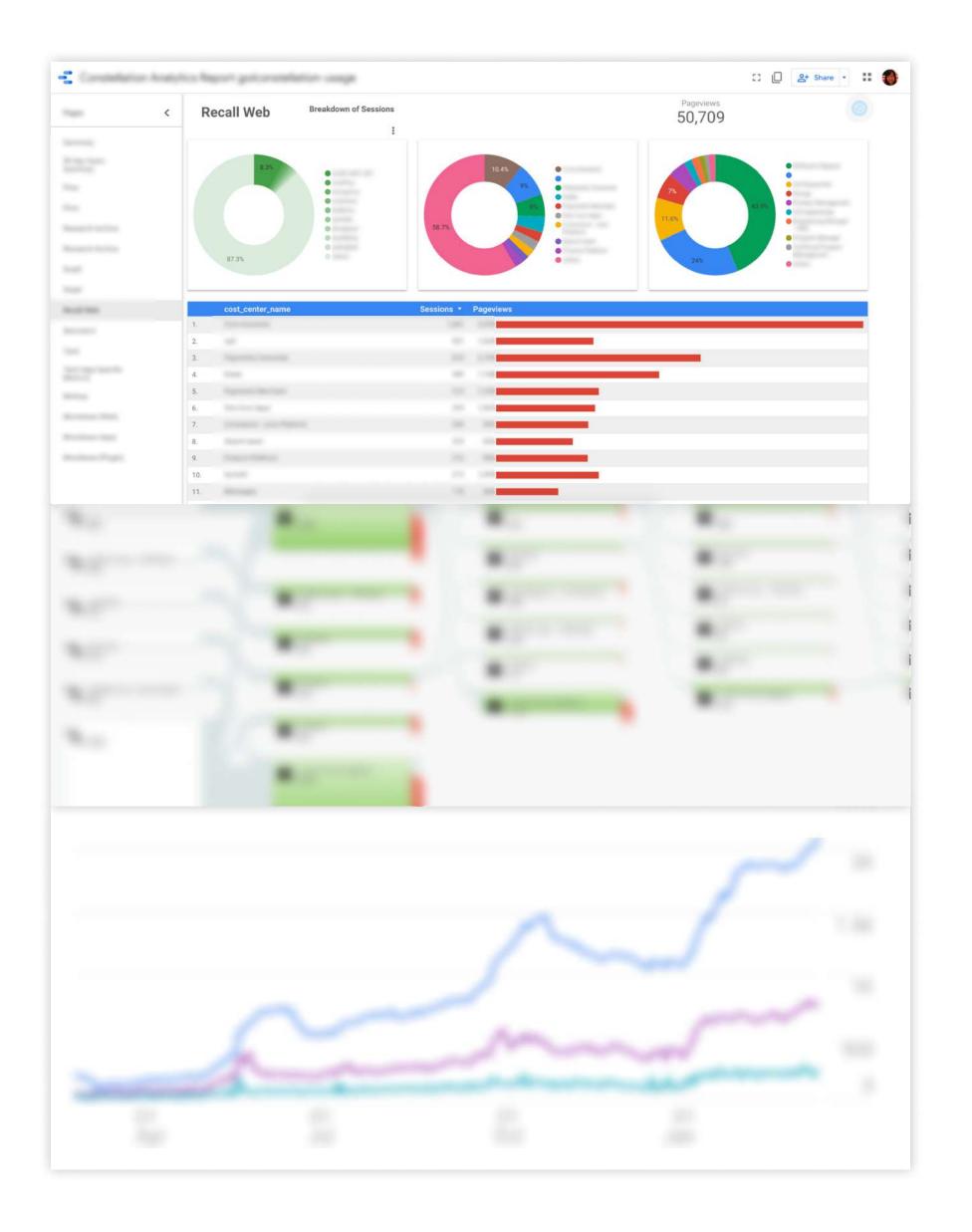
Find out users' frustrations and how Recall fits into the market of similar platforms.

Find out how we can try to capitalize on our niche to win over new users.

Refine the visual design of Recall to position ourselves as a polished product.

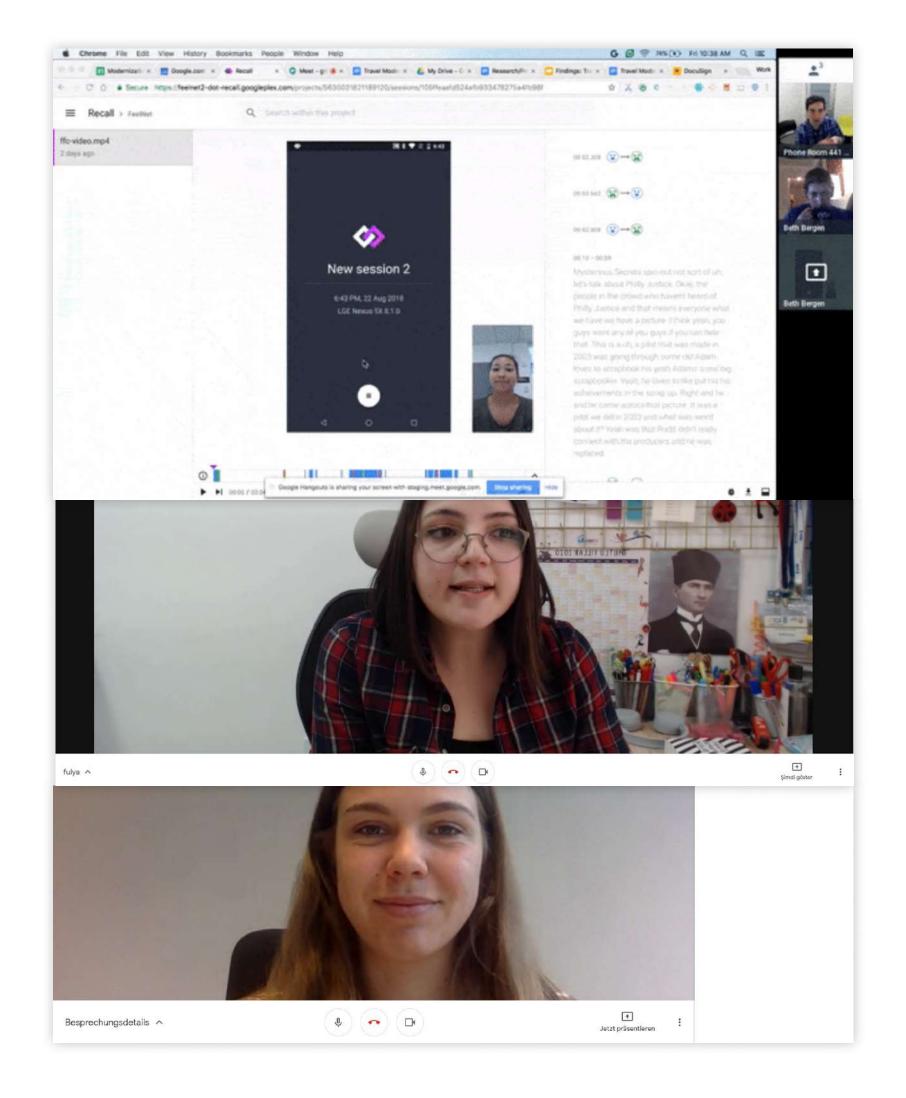
QUANTITATIVE DATA

- UX researchers actually make up less than 50% of our userbase
- The vast majority of our users have actually not created any projects, meaning low early engagement
- Significant dropoff early on in user journeys, such as the homepage



QUALITATIVE DATA

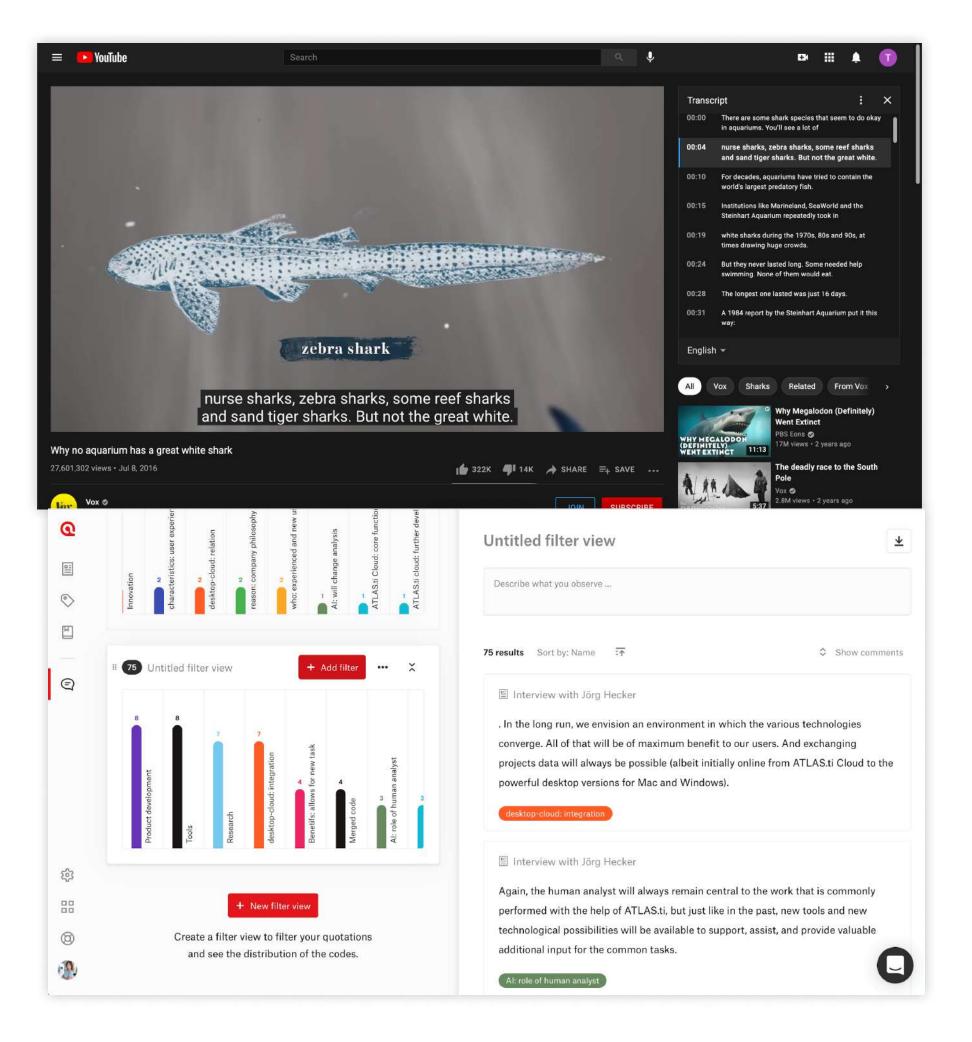
- UX researchers feel like
 Recall's analysis is error
 prone and can't be trusted
- Most interviewees do not know what most buttons and features are for
- Interviewees who weren't researchers felt that video upload errors happened too frequently and were difficult to discover and fix



14 informal interviews with users, and 14 with other UXRs.

COMPETITIVE ANALYSIS

- The vast majority of video platforms, such as Youtube or Vimeo, have almost identical Uls
- Competing software is cumbersome, with users manually entering and labeling data
- Many UX analysis apps save data to servers, making it not appropriate for use with PII/SPII



UI comparison with Youtube and flow comparison with ATLAS.ti.

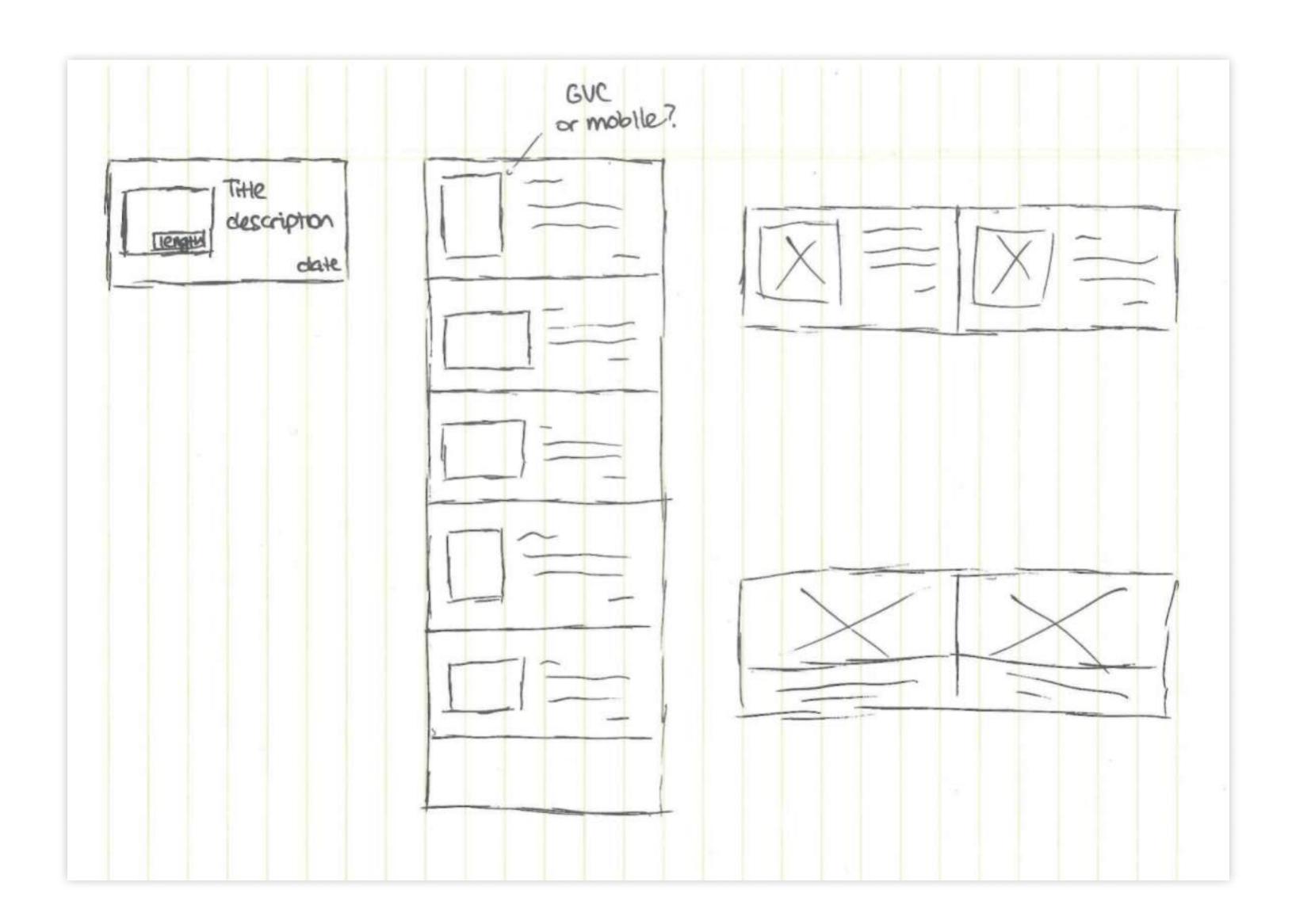
GUIDING PRINCIPLES

Allow both sets of users to access the features they prioritize

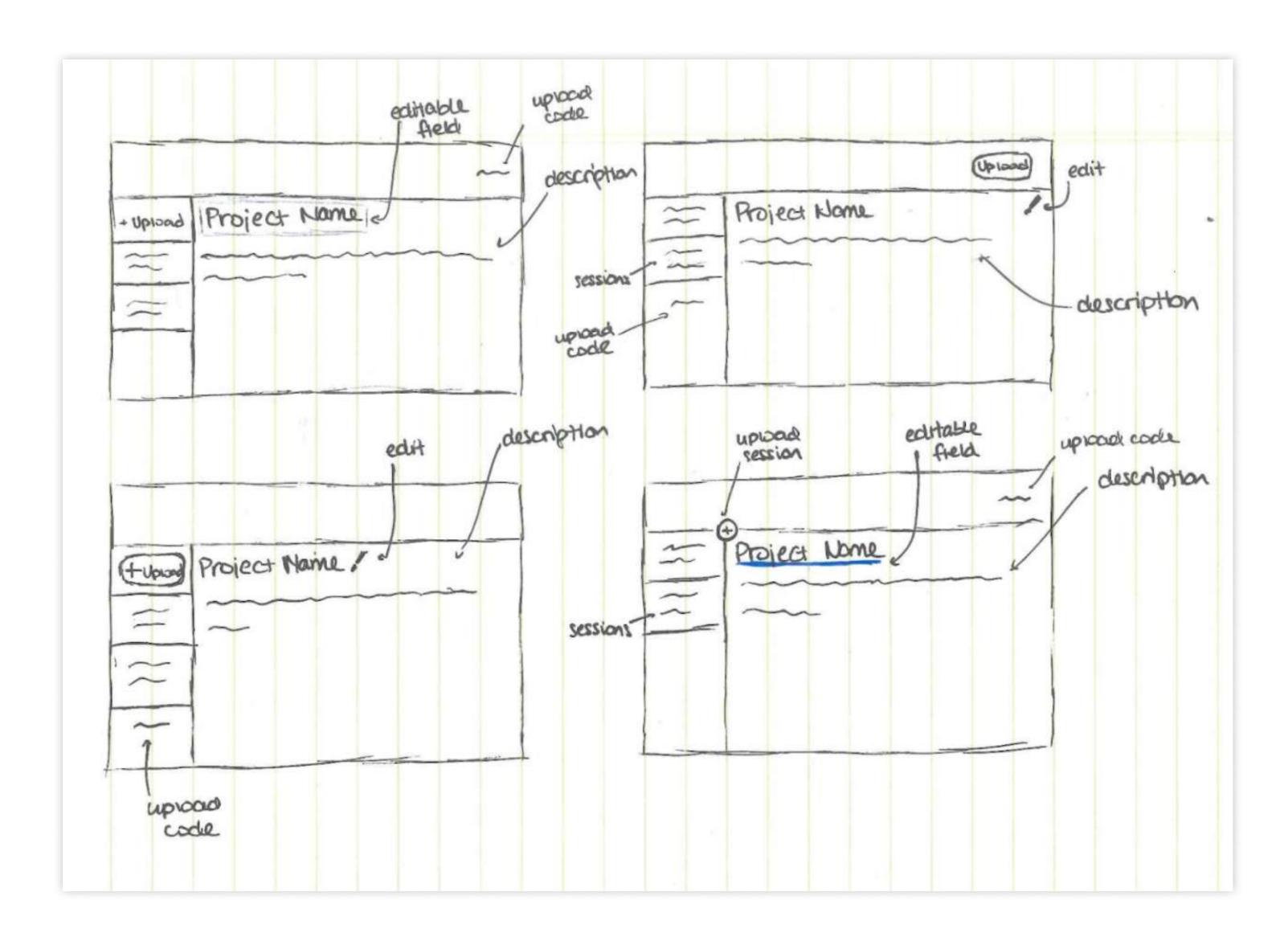
Ease users into Recall by adding onboarding UX patterns

Add error-tolerant design so that users won't feel as frustrated

Guiding principles derived from insights to keep in mind going into the designing phase.

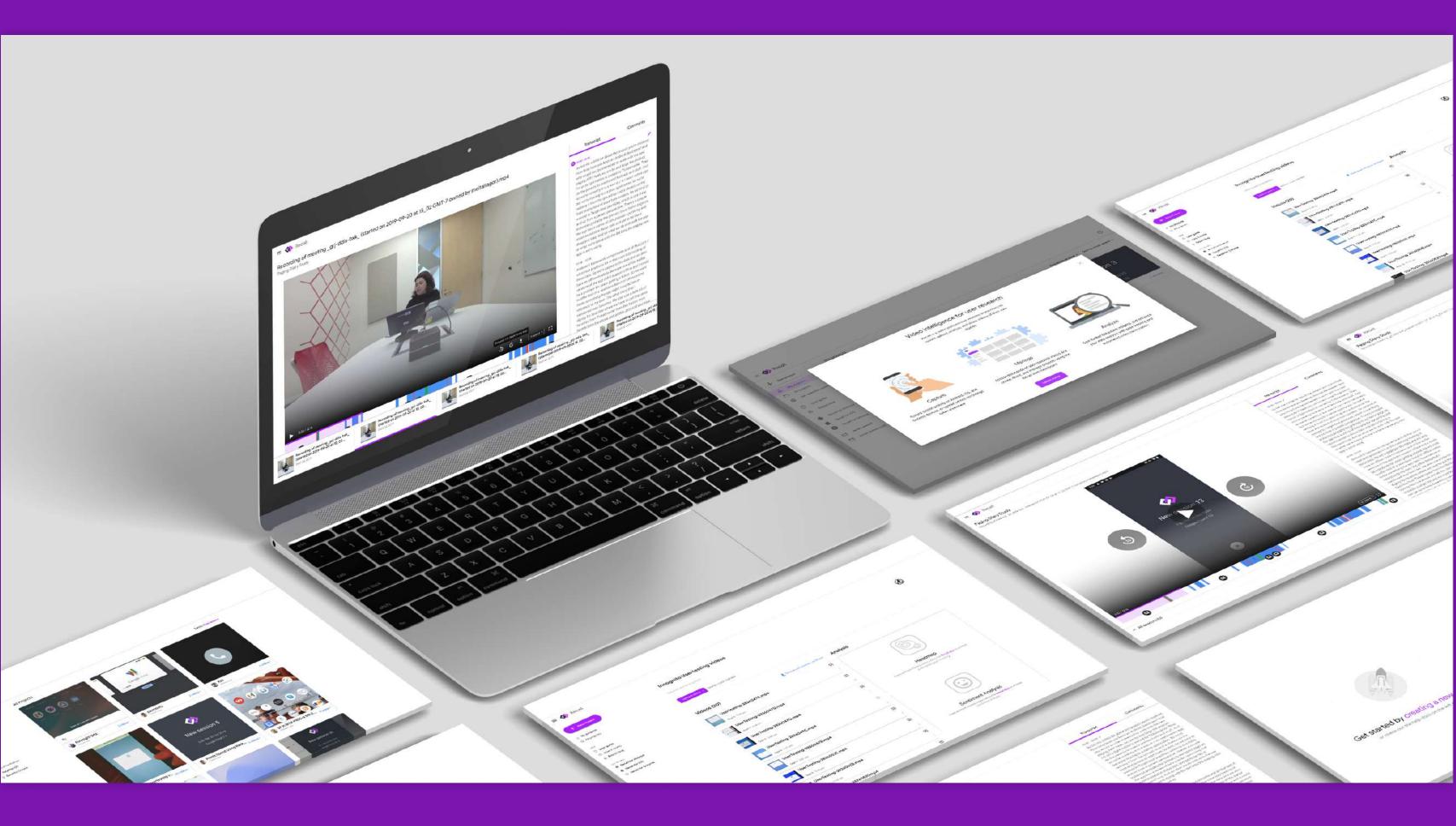


Exploring portrait and landscape thumbnails





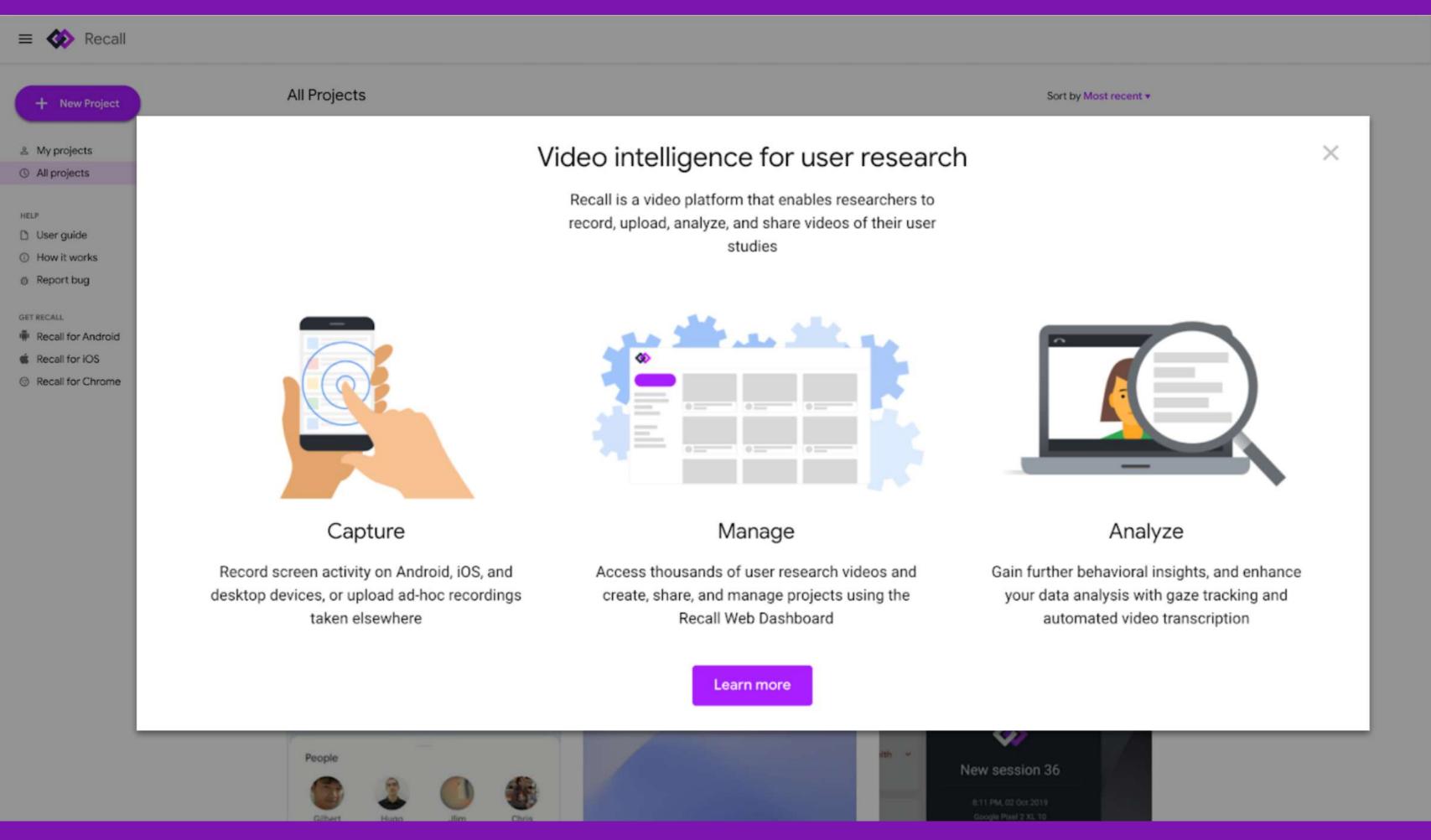
FINISHED PRODUCT



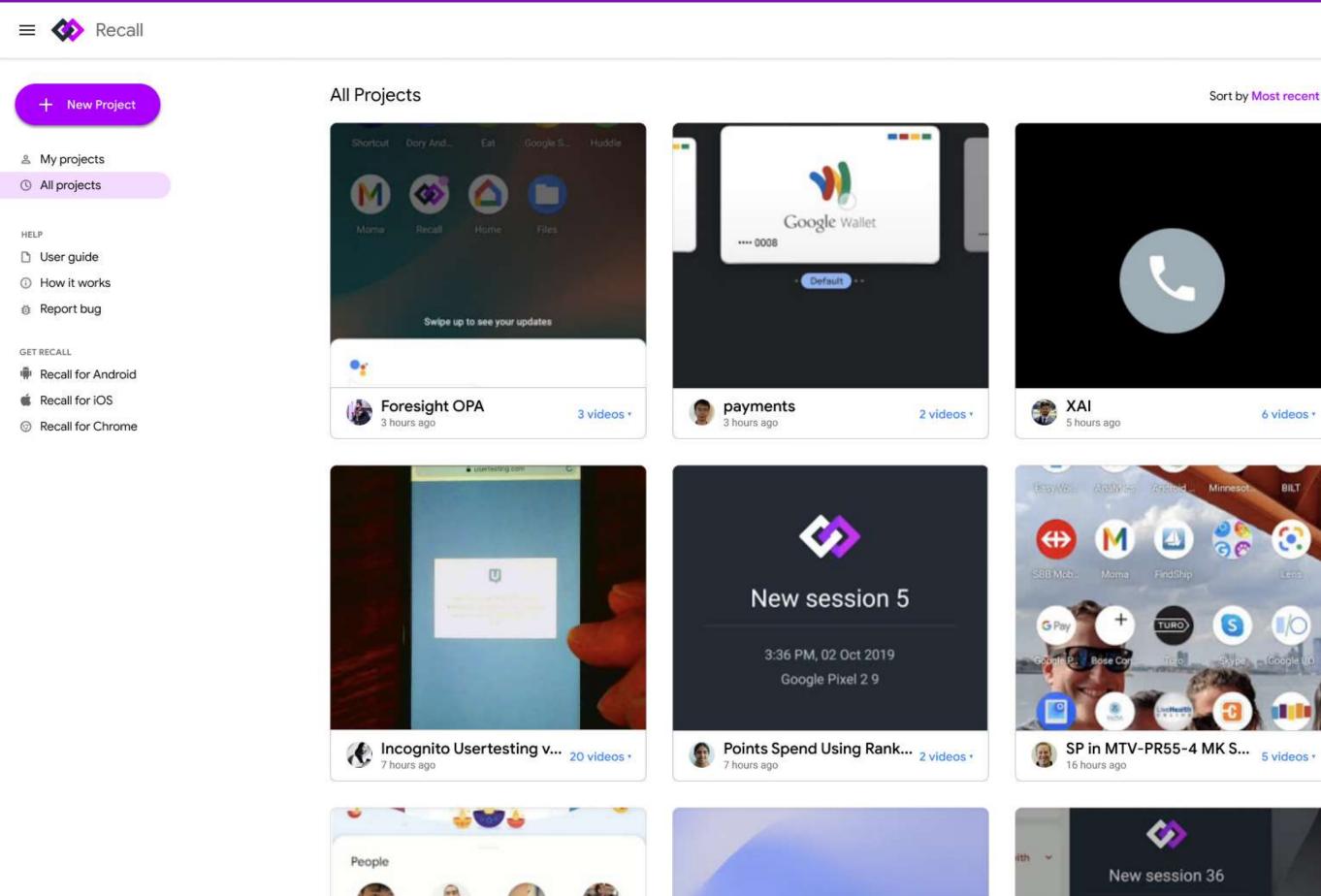
The entire process from initial ideation to final screens took about 3 months to complete.



The logo I landed on. It embodies elements of partners coming together, like research and design, while also resembling a cassette tape and recordings.



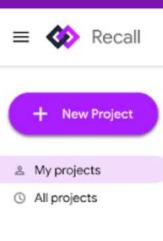
The initial screen when you first come to Recall. I created the illustrations and designed it to ease new users in.



The home page, after you close the initial tutorial. The second screen the user sees.

Sort by Most recent ▼

6 videos *



☐ User guide

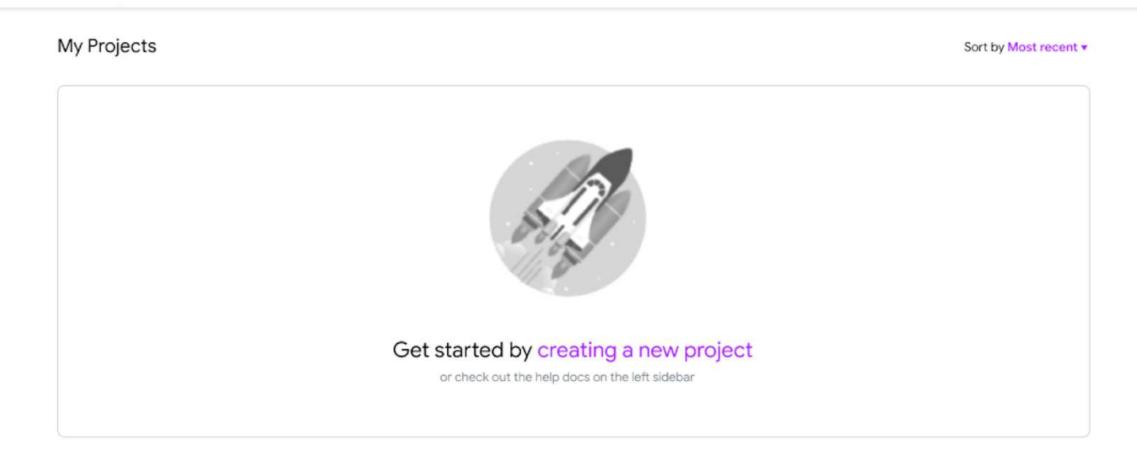
How it works

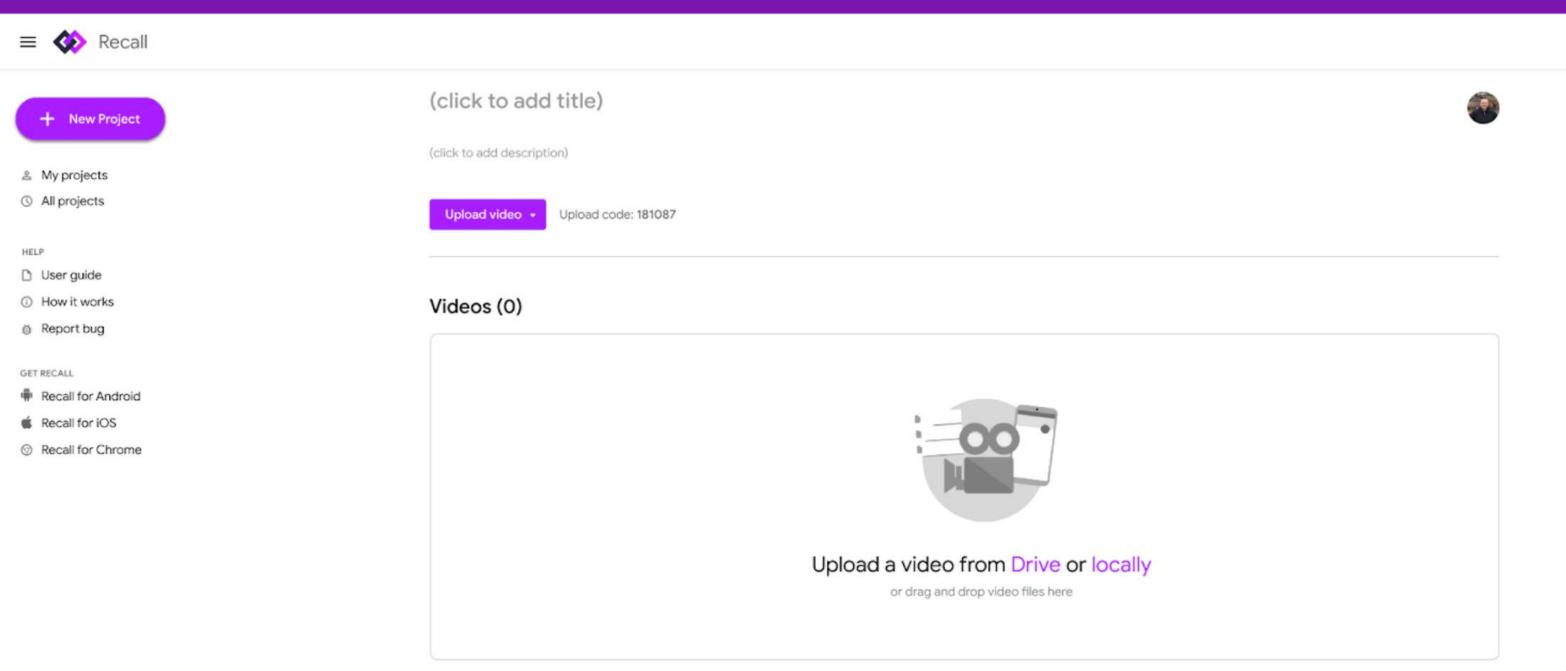
GET RECALL

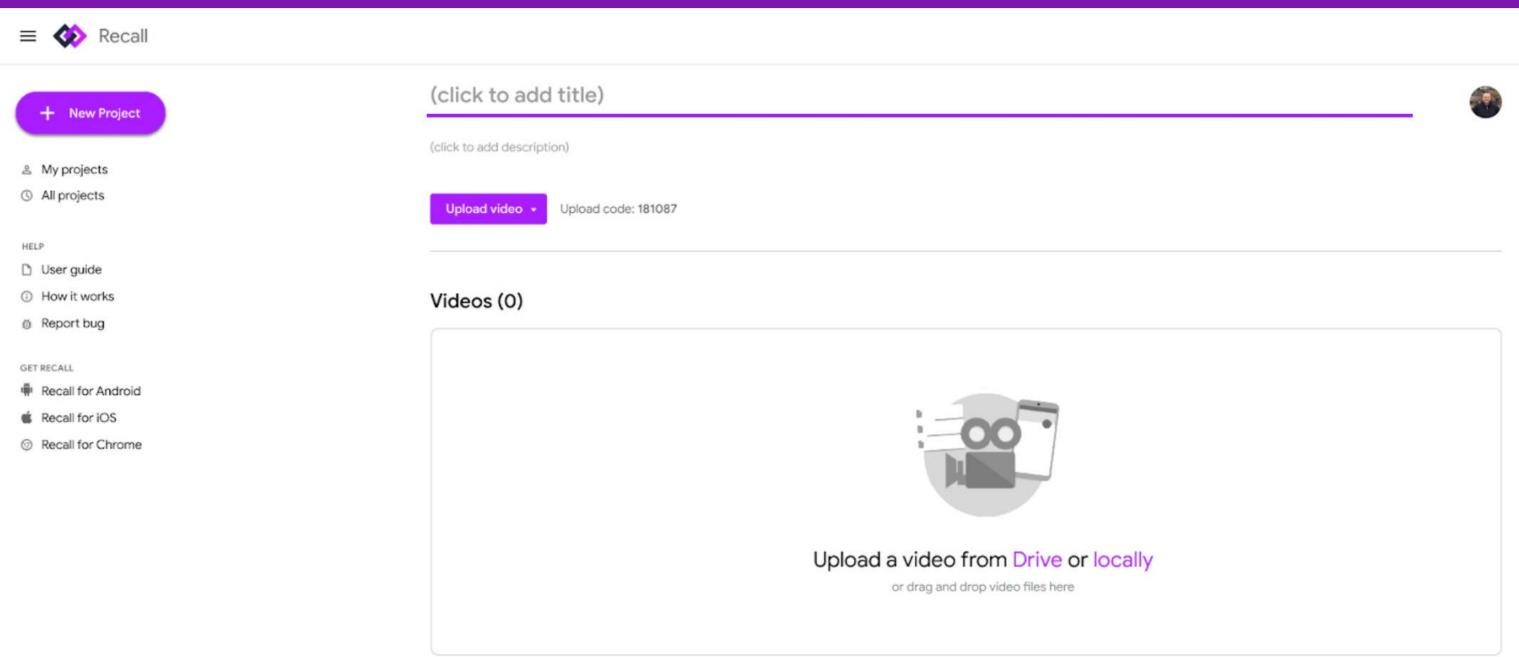
Recall for Android

Recall for iOS

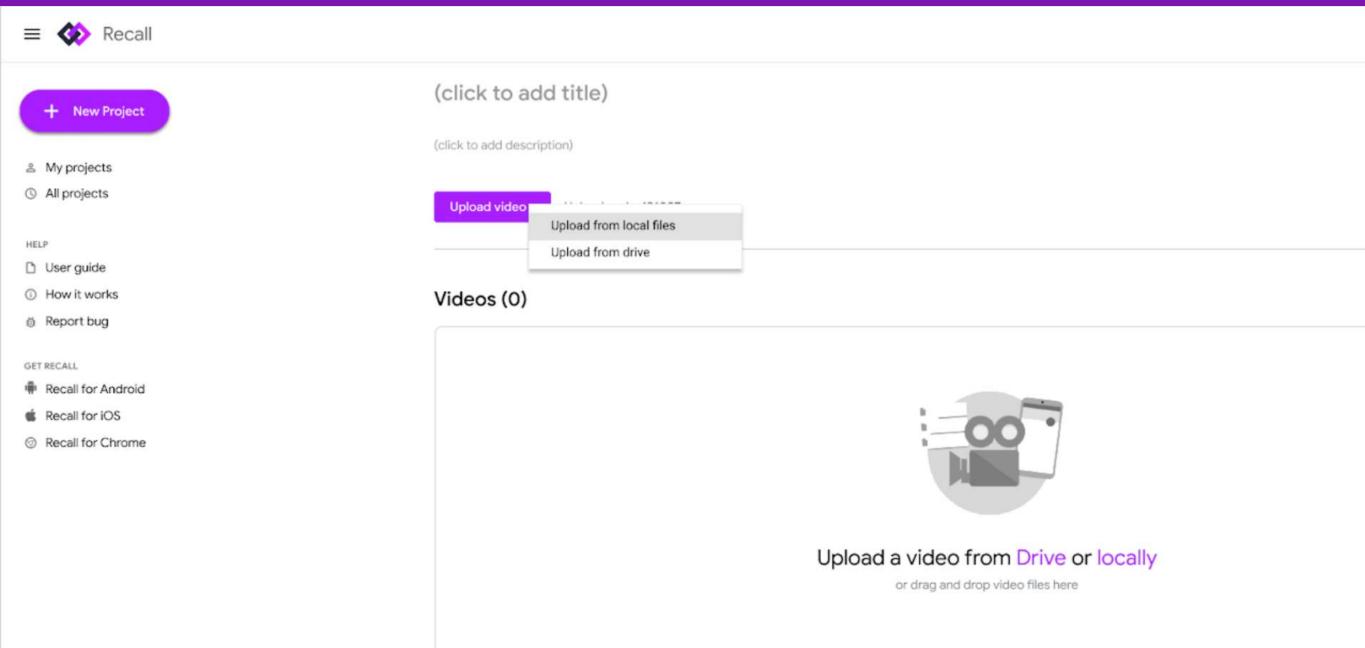
Recall for Chrome

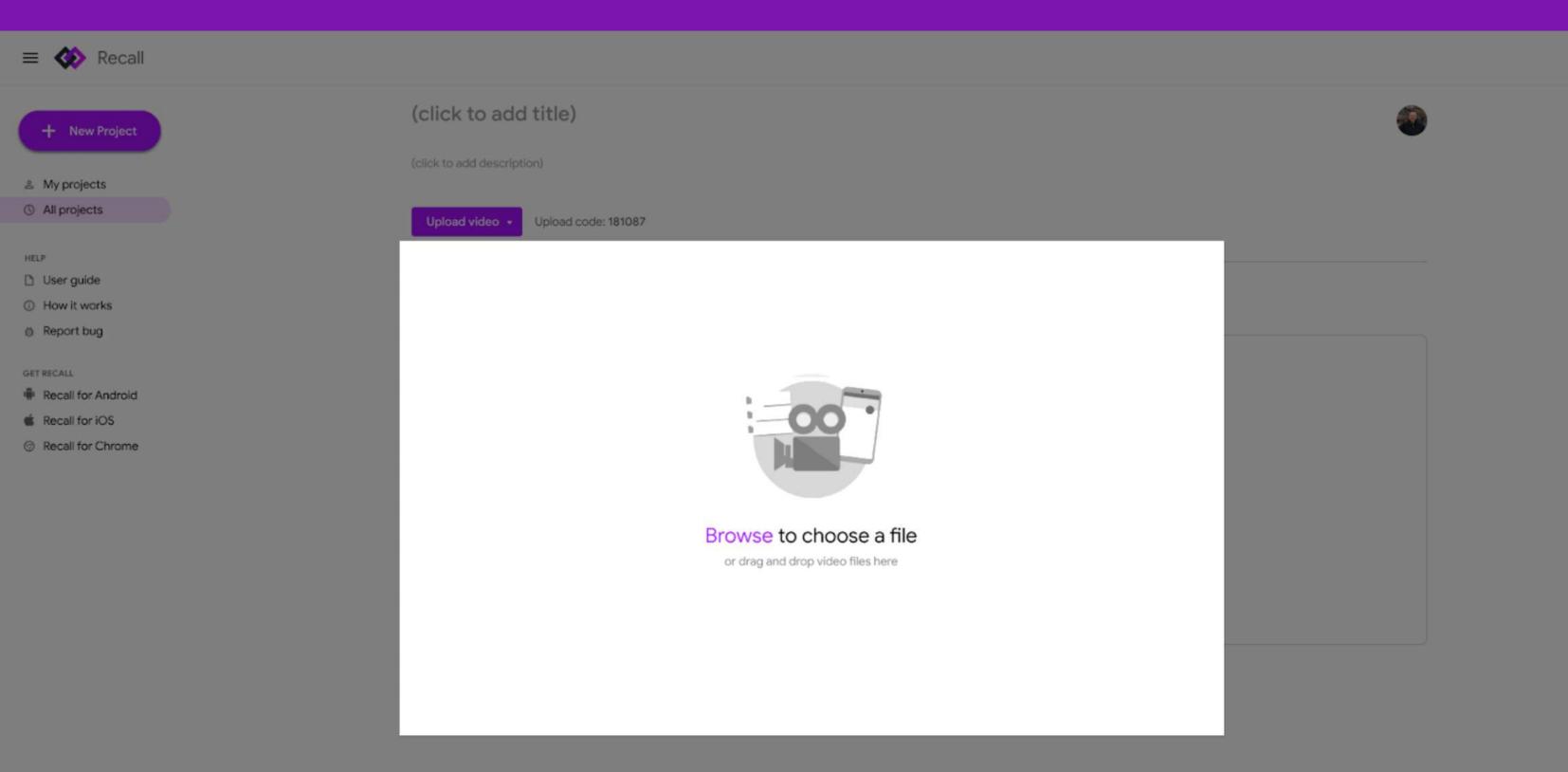


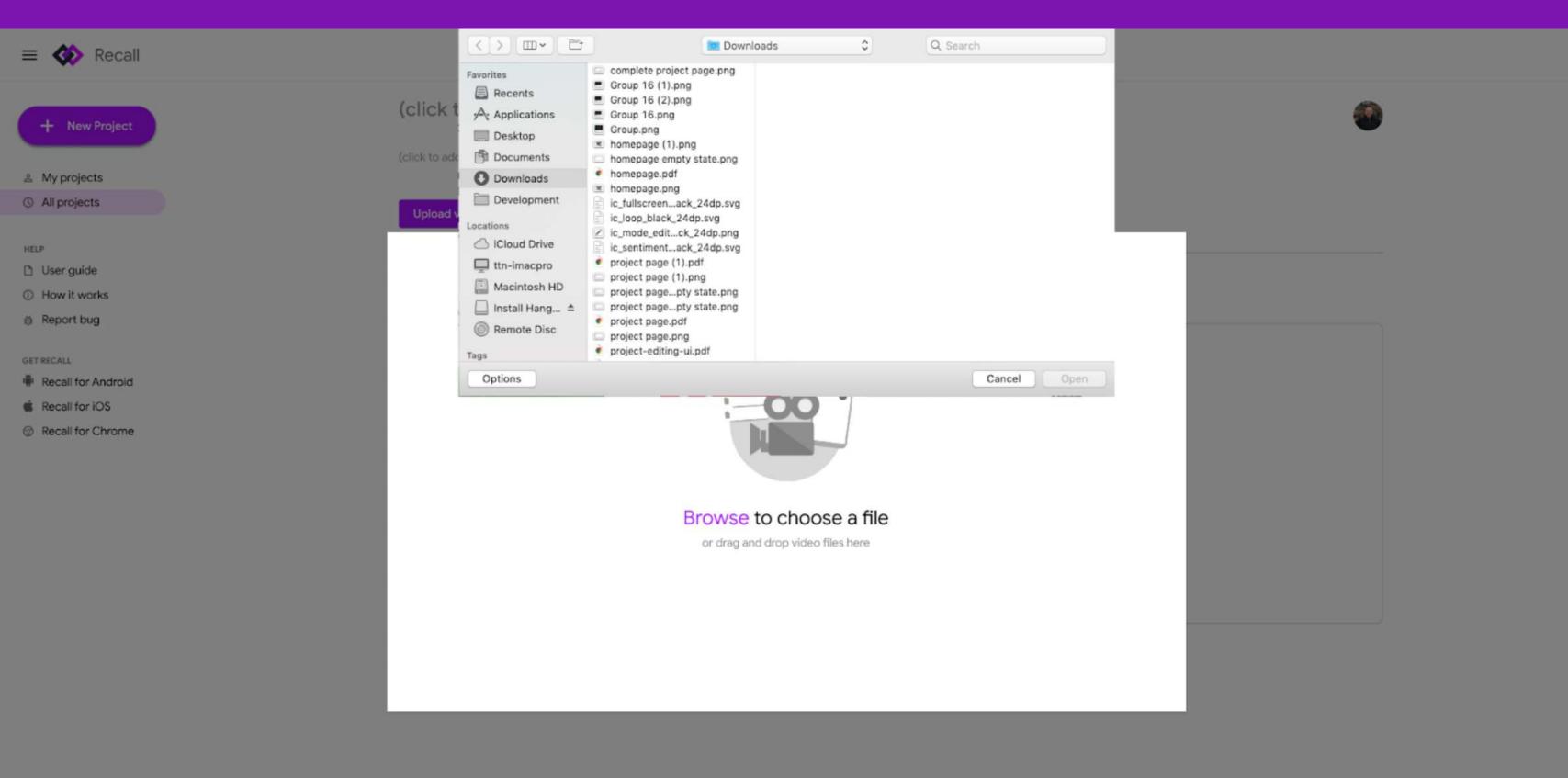


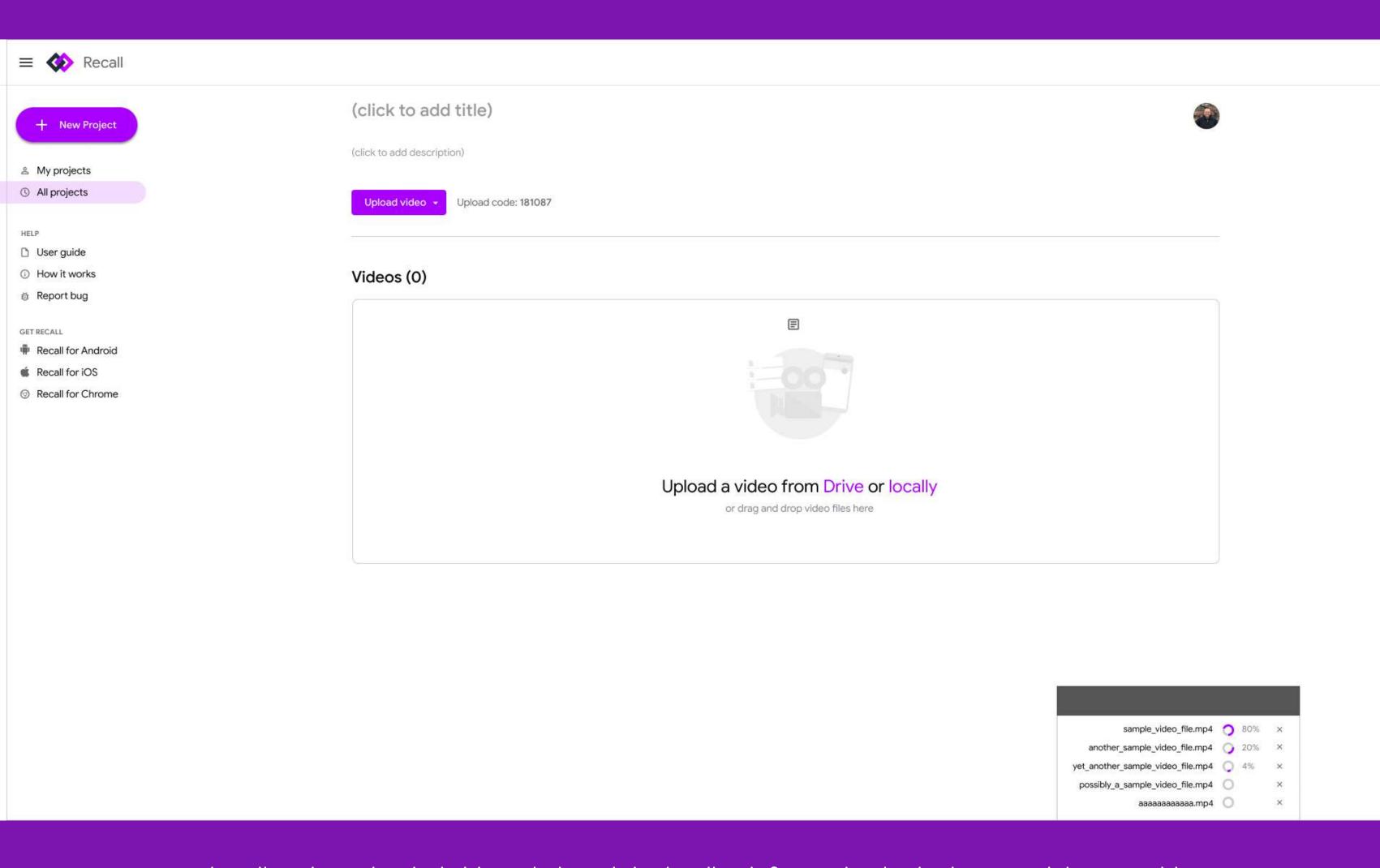


Clicking an element allows you to edit it. This allows for a layout where users have an idea of what a complete project page will look like and an intuitive understanding of how to use the page.

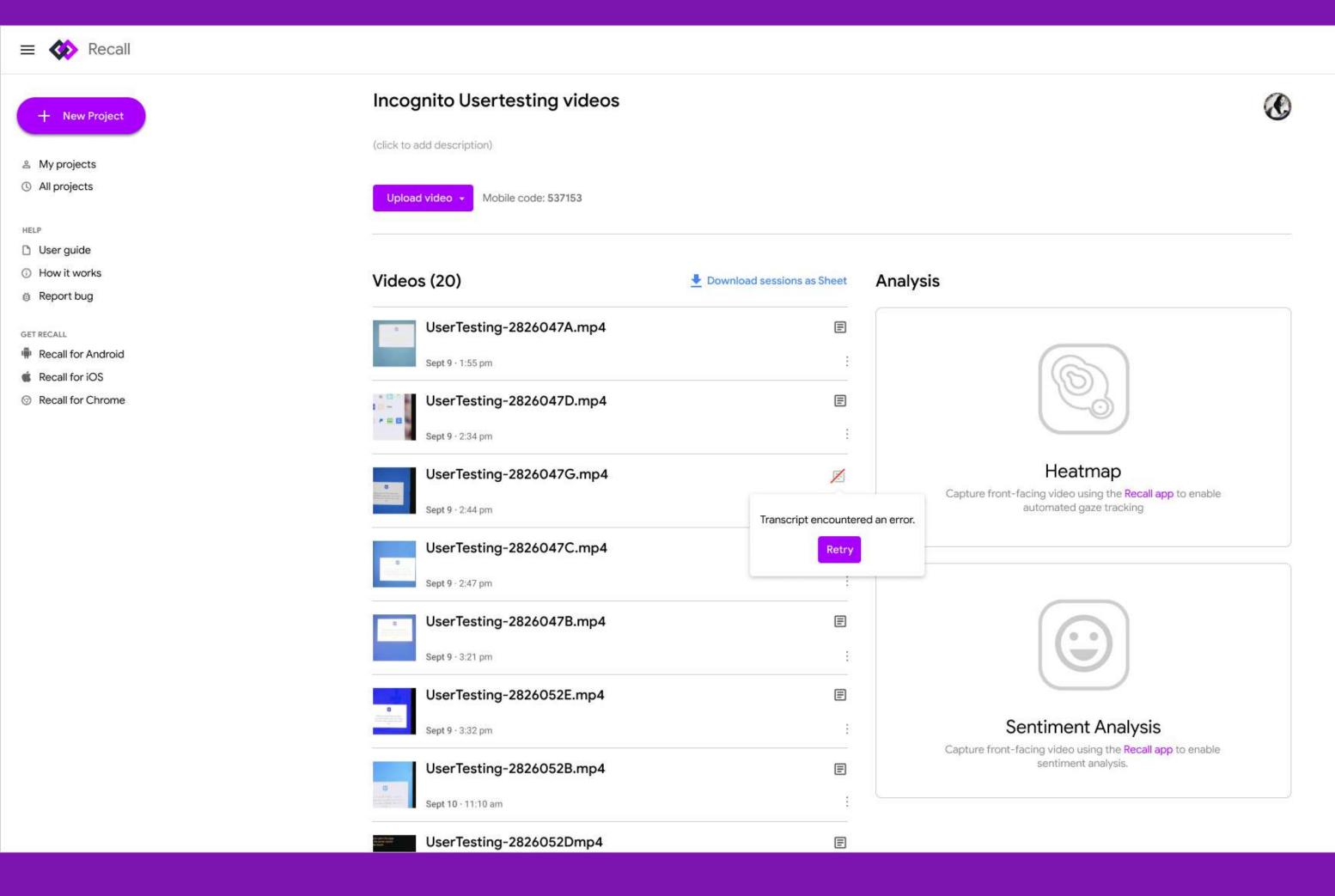




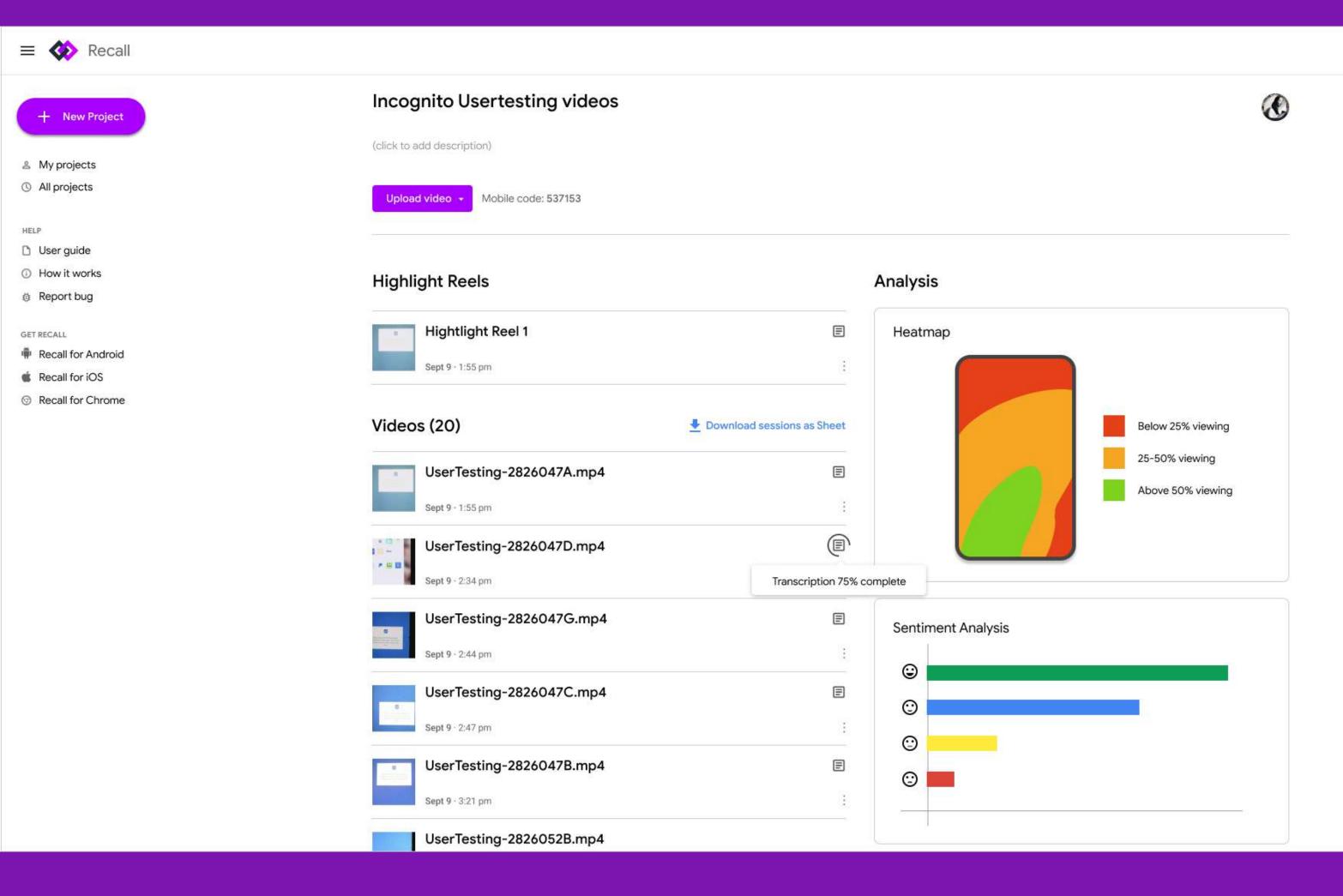




Loading the uploaded videos. I placed the loading information in the bottom right to avoid obstructing project information, which exists on the next page.



A complete project page. I added icons to let users know if their data has uploaded successfully or errored out, in which case they can retry to ease user frustration.

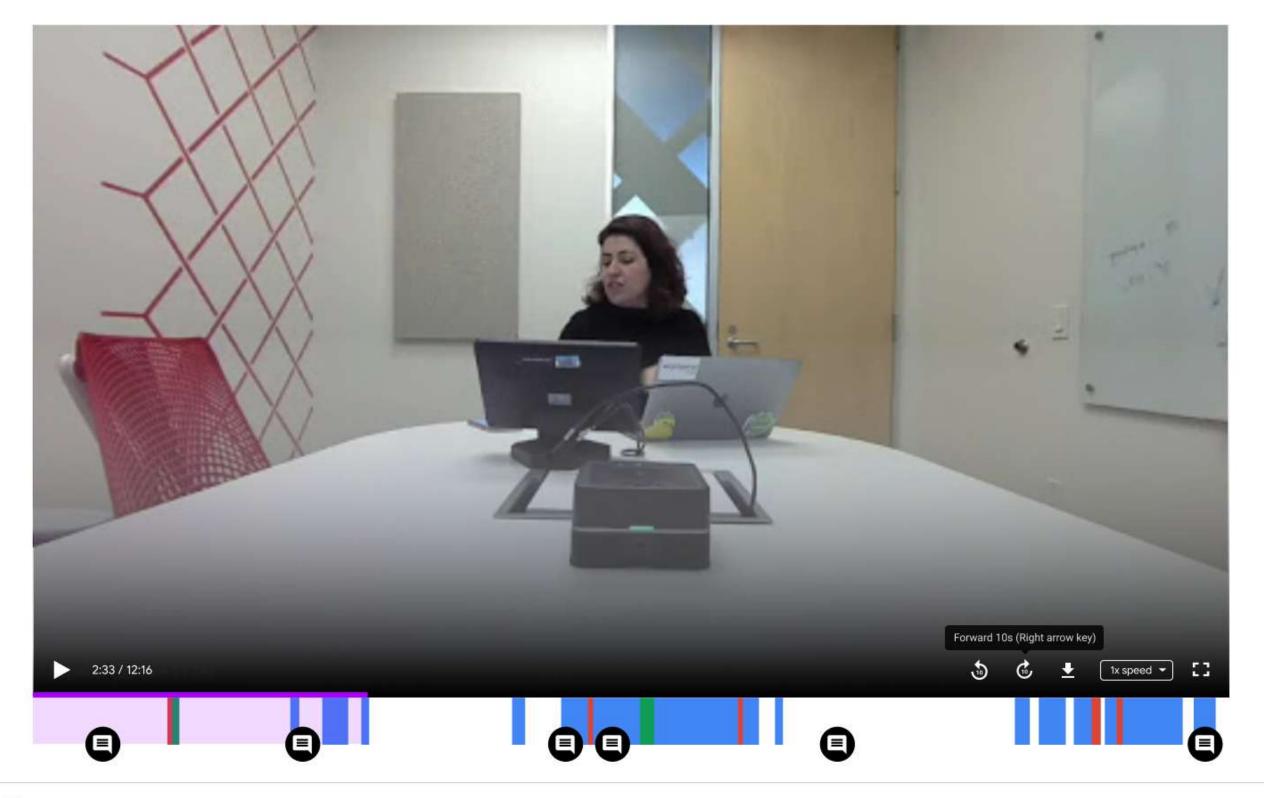


Complete project page with analysis. The analysis components resemble the empty states, in order to let new users know what the page will look like once complete.





Recording of meeting _grj-ddis-bxk_ (started on 2019-09-20 at 15_02 GMT-7 owned by meitaltagor).mp4 Paging Diary Study



Transcript

Comments



00:00 - 00:59

So tell me a little bit about the project you're involved. How long have you been an Android developer? And why would you be interested to work with the new paging API? Yeah, so, um by and large the product I'm on is light nation is owned by Ticketmaster. They do the parents to events and festivals and stuff. And so we're currently it is a was or / is I react native app, but we're moving to a native application. So we're redoing it from the ground up and a chance on the front end to kind of start from scratch. So we kind of wanted to Target best practices, which is one thing and so from a patient perspective. There's a couple instances probably the primary is our home page on the app has a series of lists popular upcoming and recommended in those lists and get to be like a thousand long. And so what we do with with the pay or what we're doing with the the beta the paging new app is we're using

01:00 - 01:59

Android X framework components and all that jazz if what best practices for in this case lazy loading of those lists. So we're paging in the data and we also have an offline mode to because one of the earlier versions of the app didn't support offline and it could be a problem for users getting a ticket. So we want to offer kind of a seamless blend of offline online mode something the app didn't have before in reference to my own. The other thing that I mentioned was favorites. We also use a little bit of dignity for favorites where we have to pull the users favorites from Ticketmaster from like Nation from Spotify from Facebook and all this Jazz and you have to create a single list and we want to display it that can get quite long depending upon some people who

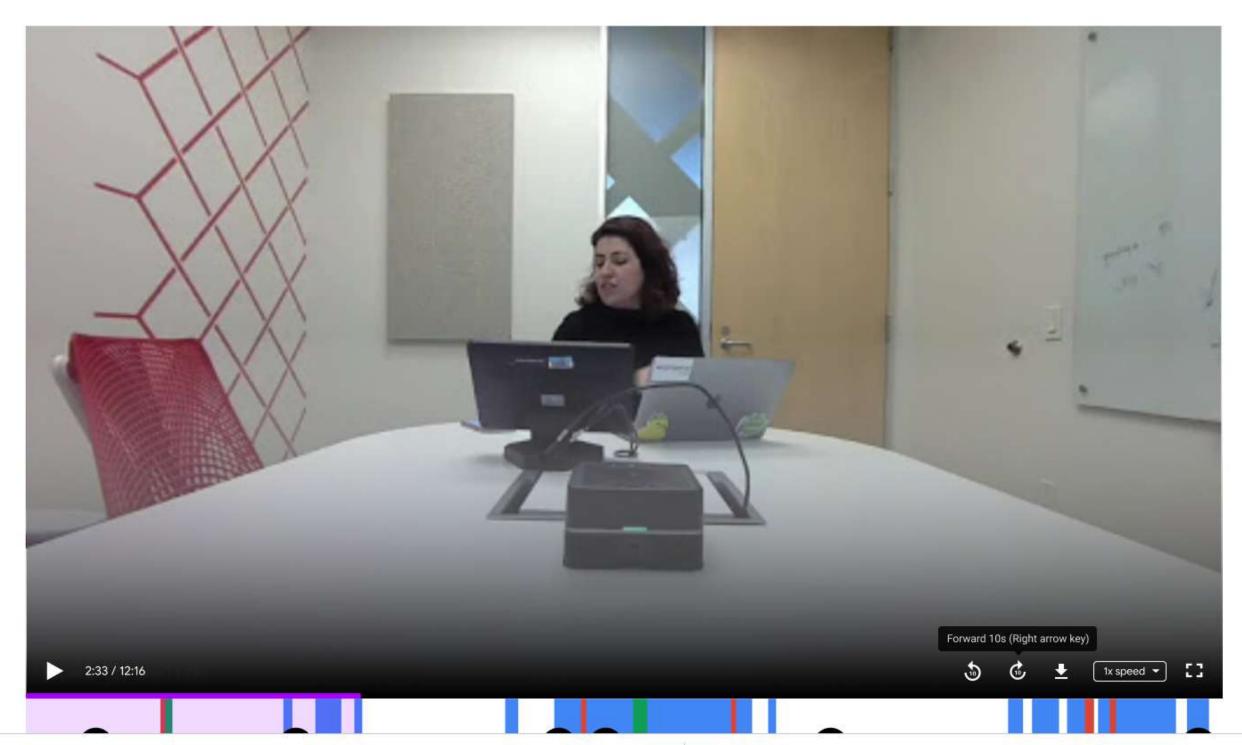
Videos (20) ∧





Recording of meeting _grj-ddis-bxk_ (started on 2019-09-20 at 15_02 GMT-7 owned by meitaltagor).mp4

Paging Diary Study



Transcript Comments



So tell me a little bit about the project you're involved. How long have you been an Android developer? And why would you be interested to work with the new paging API? Yeah, so, um by and large the product I'm on is light nation is owned by Ticketmaster. They do the parents to events and festivals and stuff. And so we're currently it is a was or / is I react native app, but we're moving to a native application. So we're redoing it from the ground up and a chance on the front end to kind of start from scratch. So we kind of wanted to Target best practices, which is one thing and so from a patient perspective. There's a couple instances probably the primary is our home page on the app has a series of lists popular upcoming and recommended in those lists and get to be like a thousand long. And so what we do with with the pay or what we're doing with the the beta the paging new app is we're using

01:00 - 01:59

Android X framework components and all that jazz if what best practices for in this case lazy loading of those lists. So we're paging in the data and we also have an offline mode to because one of the earlier versions of the app didn't support offline and it could be a problem for users getting a ticket. So we want to offer kind of a seamless blend of offline online mode something the app didn't have before in reference to my own. The other thing that I mentioned was favorites. We also use a little bit of dignity for favorites where we have to pull the users favorites from Ticketmaster from like Nation from Spotify from Facebook and all this Jazz and you have



Recording of meeting _grj-ddis-bxk_ (started on 2019-09-20 at 15 02... Sept 22, 2019



Recording of meeting _grj-ddis-bxk_ (started on 2019-09-20 at 15_02... Sept 22, 2019



Recording of meeting _grj-ddis-bxk_ (started on 2019-09-20 at 15_02... Sept 22, 2019



Recording of meeting _grj-ddis-bxk_ (started on 2019-09-20 at 15_02... Sept 22, 2019



Recording of meeting _grj-dd (started on 2019-09-20 at 15_ Sept 22, 2019

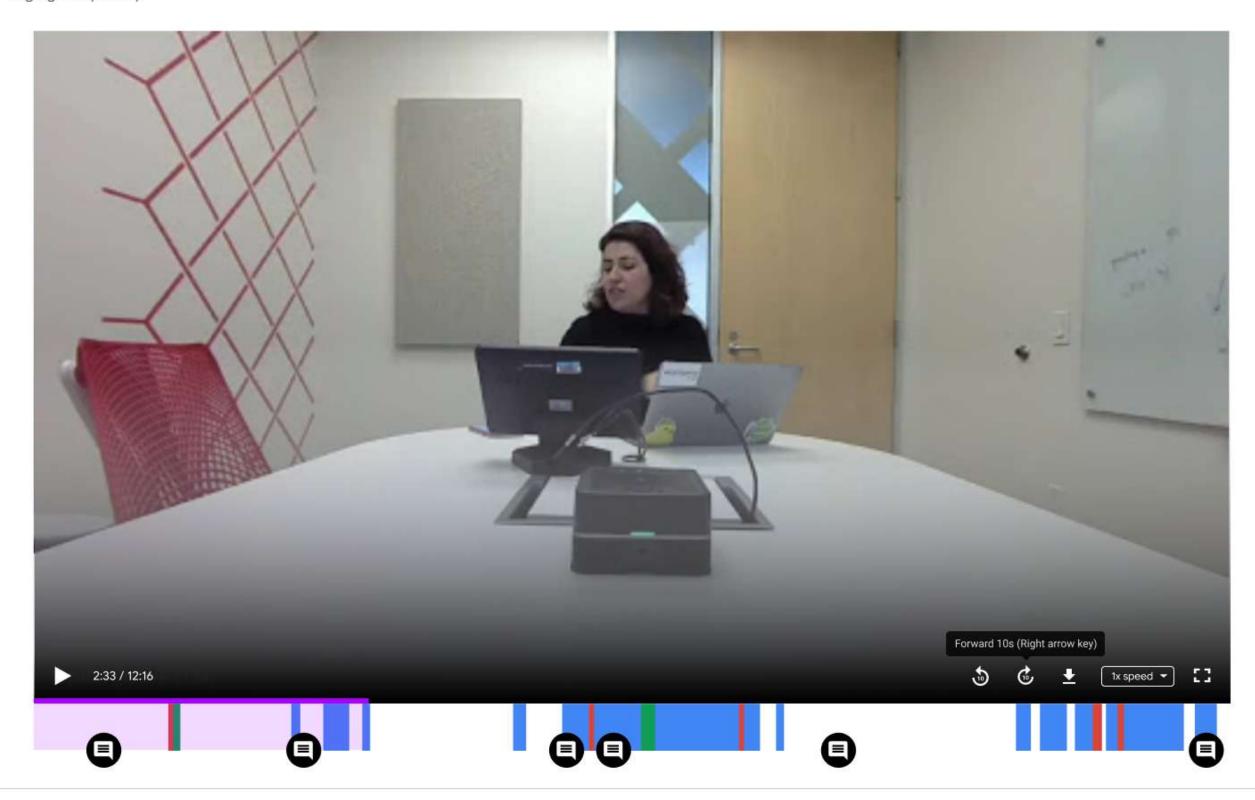
Opening up the video list at the bottom. Because I knew internal users would be using a Macbook or Chromebook, I was able to implement a horizontal scrolling design to efficiently use space.





Recording of meeting _grj-ddis-bxk_ (started on 2019-09-20 at 15_02 GMT-7 owned by meitaltagor).mp4

Paging Diary Study



Transcript

Comments

00:00 - 00:59

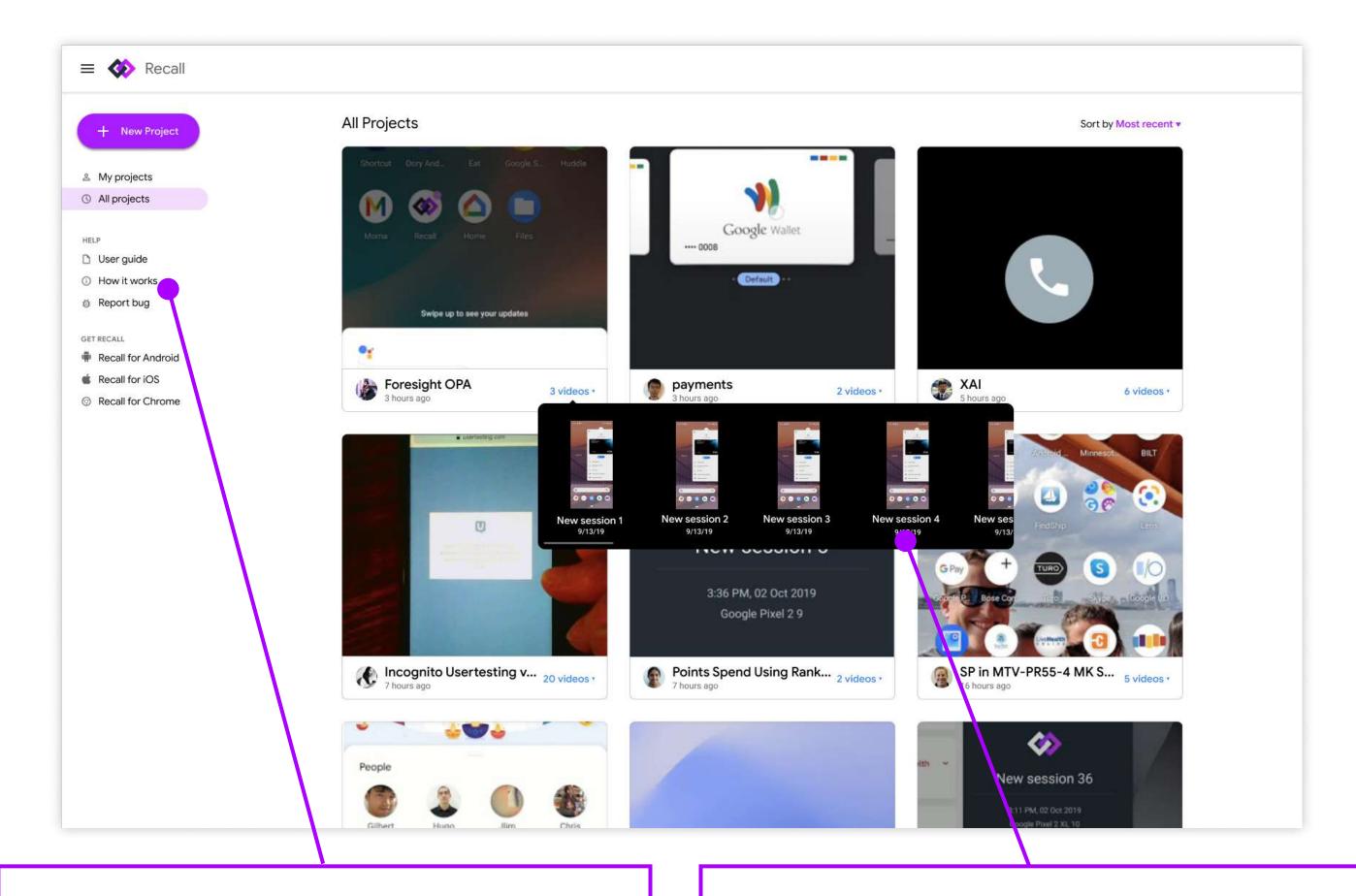
So tell me a little bit about the project you're involved. How long have you been an Android developer? And why would you be interested to work with the new paging API? Yeah, so, um by and large the product I'm creating this for is owned by Ticketmaster. They do the parents to events and festivals and stuff. And so we're currently it is a was or / is I react native app, but we're moving to a native application. So we're redoing it from the ground up and a chance on the front end to kind of start from scratch. So we kind of wanted to Target best practices, which is one thing and so from a patient perspective. There's a couple instances probably the primary is our home page on the app has a series of lists popular upcoming and recommended in those lists and get to be like a thousand long. And so what we do with with the pay or what we're doing with the the beta the paging new app is we're using

01:00 - 01:59

Android X framework components and all that jazz if what best practices for in this case lazy loading of those lists. So we're paging in the data and we also have an offline mode to because one of the earlier versions of the app didn't support offline and it could be a problem for users getting a ticket. So we want to offer kind of a seamless blend of offline online mode something the app didn't have before in reference to my own. The other thing that I mentioned was favorites. We also use a little bit of dignity for favorites where we have to pull the users favorites from Ticketmaster from like Nation from Spotify from Facebook and all this Jazz and you have to create a single list and we want to display it that can get quite long depending upon some people who

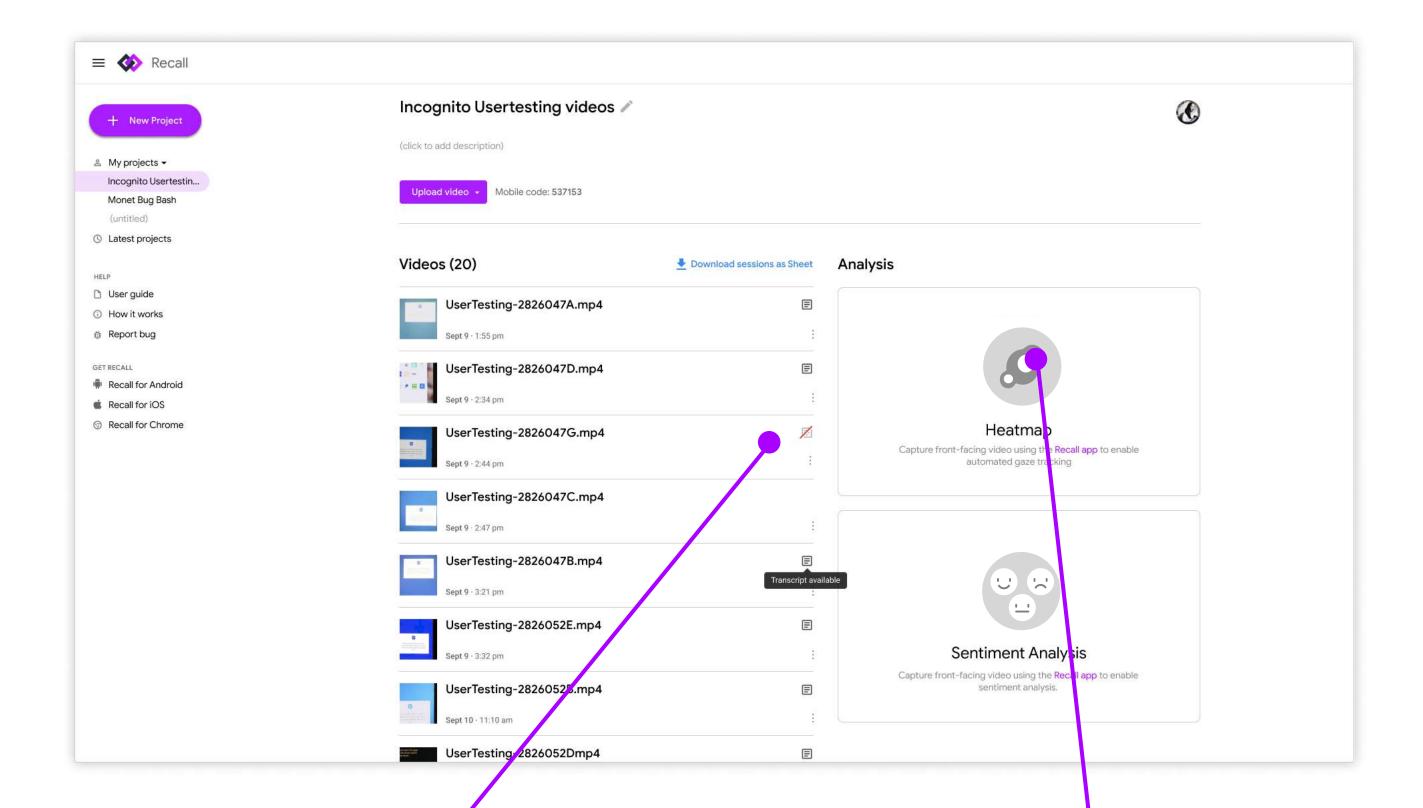
| Videos (20) ∧

I added transcript editing on the right side, so that users who faced transcription errors could fix it themselves and work towards a correct transcript, which solved a huge pain point.



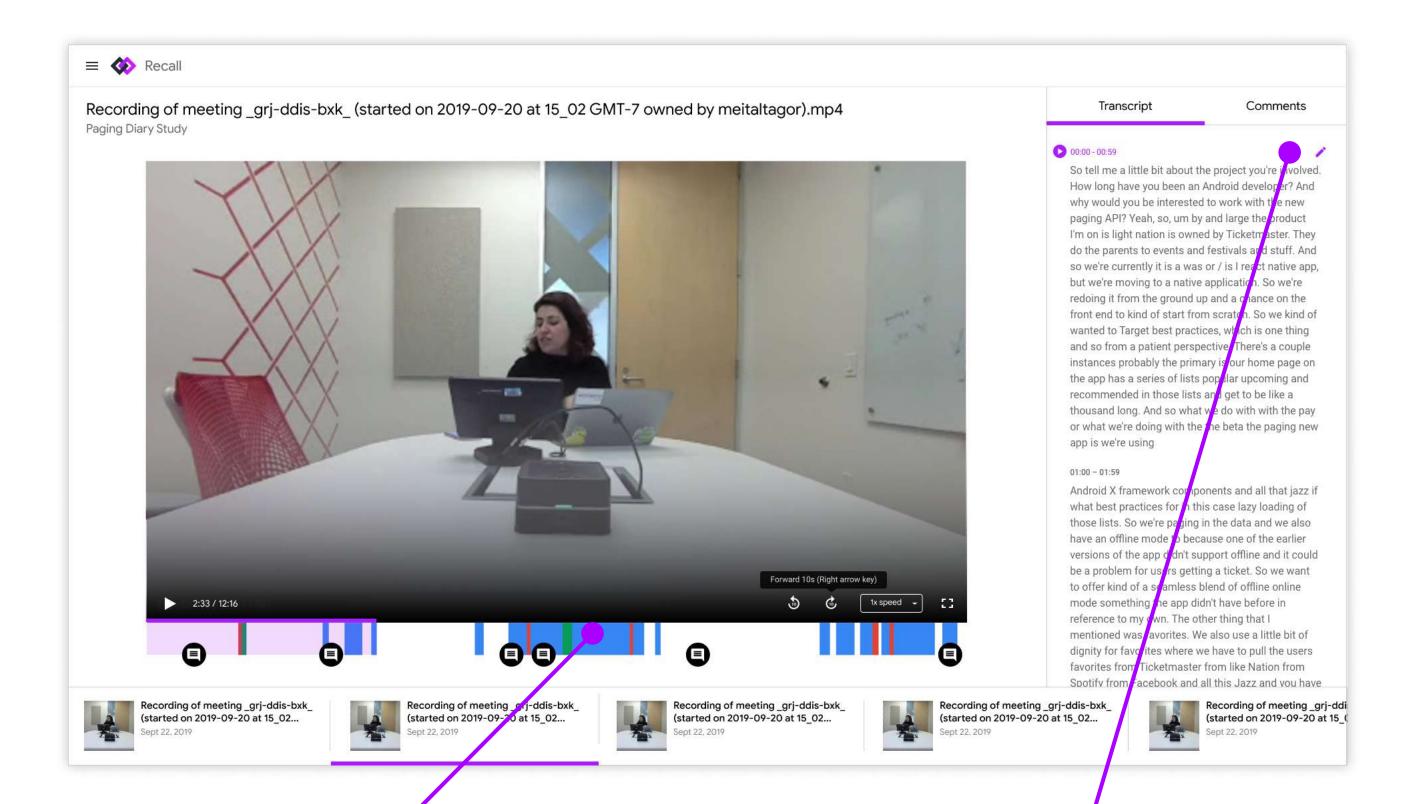
I found that new users felt disoriented and couldn't find features landing on the homepage for the first time, so I made sure the user guide was surfaced at all times so that they could be eased into the product.

I learned that engineers wanted to keep their flow short and simple, so I added the ability to jump to an individual video. This allows researchers to access project-level metrics while not burdering engineers' flow.



I added icons to let users know at a glance if their automated analysis is in progress, completed, or failed. Users can retry on this screen to reduce frustration.

I created placeholder illustrations as empty states for a complete page to let new users know what functionality exists and how to access it.

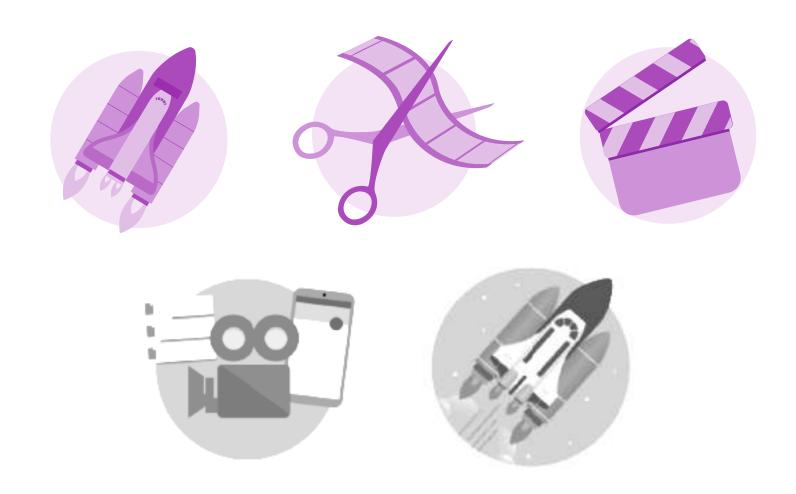


I found that researchers were primarily interested in qualitative analysis, so I added a sentiment bar beneath the video so users can immediately see what the dominant sentiment is.

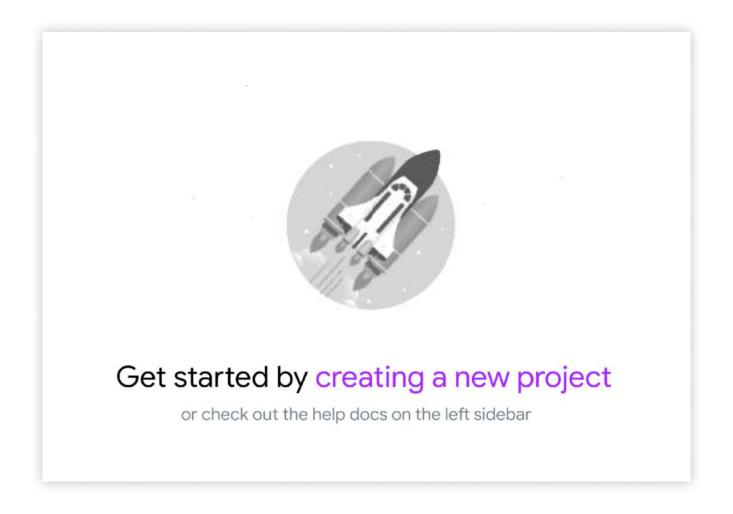
I discovered that researchers felt that the transcriptions were unreliable and unfixable, so I added transcript editing functionality to ease their frustrations.

ILLUSTRATION

I made my own illustrations, extended from the standard Google illustration style.



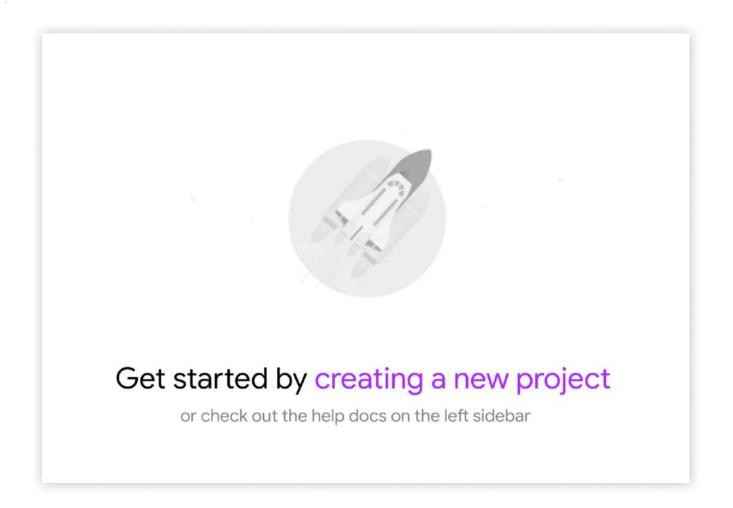




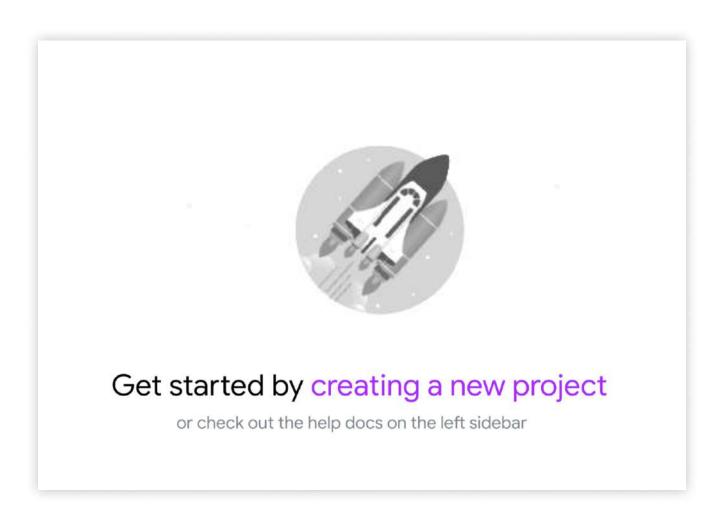
Example illustration usage to complement empty state text

I did a full ally audit of Recall to make sure everything was colorblind accessible and WCAG AA level accessible.

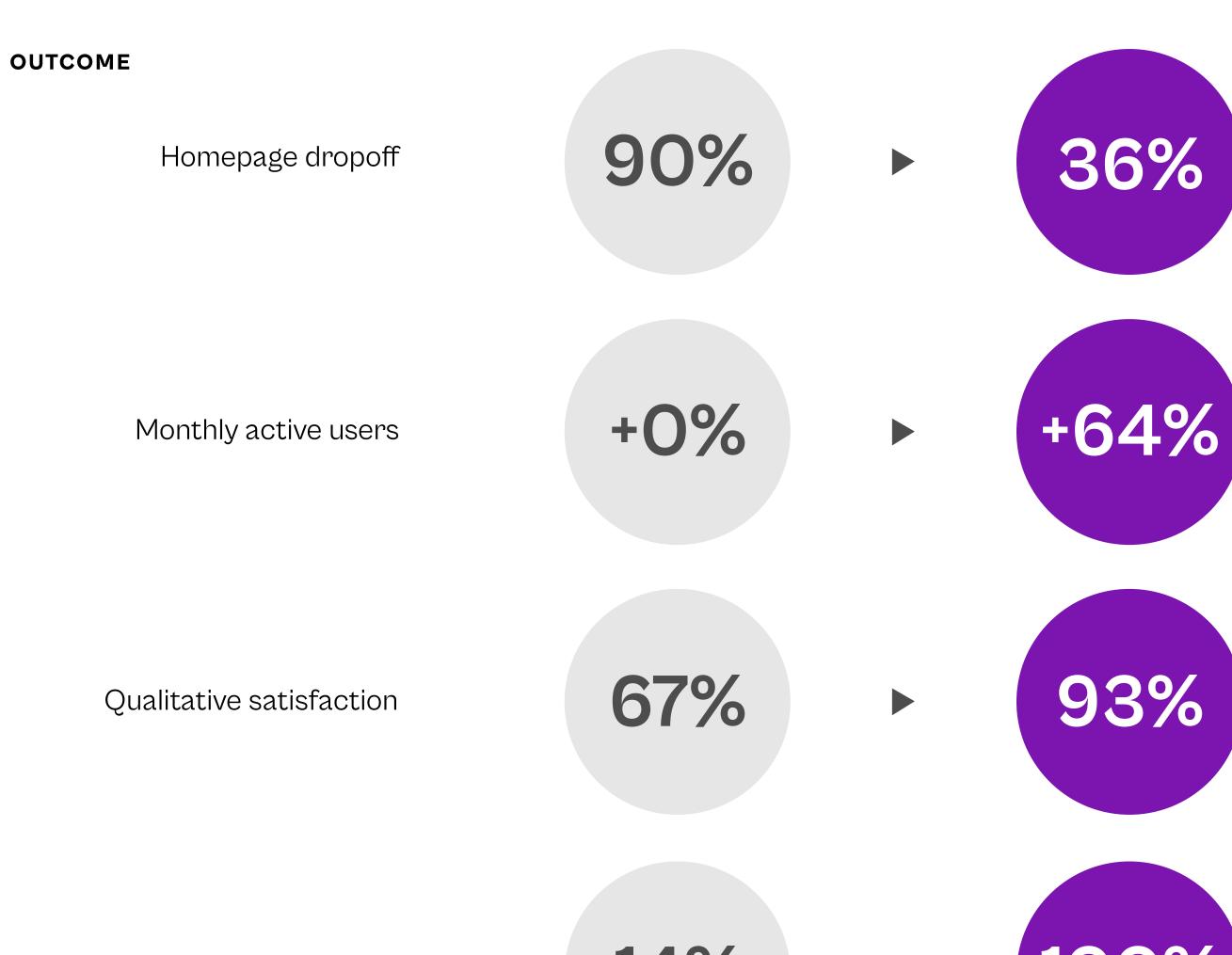
Fails AA level test



AA level accessibility



Contrast ratio: 1.16:1 Contrast ratio: 4.94:1



Journey completion

14%

100%

Your beautiful work elevated our branding and creativity and will hopefully lead to higher engagement.

We sincerely appreciate your willingness to jump right in, conceptualize, and iterate with us.

